

# AudioPlayer Interface Reference

- [Introduction](#)
- [AudioPlayer Directives](#)
- [AudioPlayer Requests](#)
- [Play Directive](#)
- [Stop Directive](#)
- [ClearQueue Directive](#)
- [PlaybackStarted Request](#)
- [PlaybackFinished Request](#)
- [PlaybackStopped Request](#)
- [PlaybackNearlyFinished Request](#)
- [PlaybackFailed Request](#)
- [System.ExceptionEncountered Request](#)
- [Other Reference Materials](#)

## Introduction

The AudioPlayer interface provides directives and requests for streaming audio and monitoring playback progression. Your skill can send *directives* to start and stop the playback. The Alexa service sends your skill

AudioPlayer requests to give you information about the playback state, such as when the track is nearly finished, or when playback starts and stops. Alexa also sends *PlaybackController requests* in response to hardware buttons such as on a remote control.

**Note:** Fire TV devices do not currently support the AudioPlayer interface.

## Configuring Your Skill for the AudioPlayer Directives

When using these directives for audio playback, you must do the following:

- Indicate that your skill implements this interface when configuring the skill. On the **Skill Information** page in the developer portal, set the **Audio Player** option to **Yes**.
- Include the required built-in intents for pausing and resuming audio in your intent schema and implement them in some way:
  - `AMAZON.PauseIntent`
  - `AMAZON.ResumeIntent`

In addition to the required built-in intents, your skill should gracefully handle the following additional built-in intents:

- `AMAZON.CancelIntent`

## Alexa Skills Kit - Custom Skills

- + [Understanding Custom Skills](#)
- + [Get Started with Sample Code](#)
- + [Voice Design](#)
- + [Setting Up](#)
- + [Developing](#)
- + [Displaying Skill Cards](#)
- + [Testing](#)
- + [Publishing](#)
- [Reference](#)
  - [Service Interface Reference \(JSON\)](#)
  - [JSON Interface Reference for Custom Skills](#)
  - [Standard Request Types Reference](#)
  - [AudioPlayer Interface Reference](#)
  - [PlaybackController Interface Reference](#)
  - [Interaction Model Reference \(Intents, Slots, Sample Utterances\)](#)
  - + [Built-in Library](#)
  - [Custom Skill Java Library Reference](#)
  - + [Speech Synthesis Markup Language \(SSML\) Reference](#)
  - [Alexa Skills English \(UK\) Migration FAQ](#)
- Other Resources**
  - [Getting Started](#)
  - [Smart Home Skills](#)
  - [Flash Briefing Skills](#)
  - [Glossary](#)
  - [Alexa Skills Kit Forum](#)
  - [Alexa Voice Service](#)



- AMAZON.NextIntent
- AMAZON.PreviousIntent
- AMAZON.RepeatIntent
- AMAZON.ShuffleOffIntent
- AMAZON.ShuffleOnIntent
- AMAZON.StartOverIntent

Note that users can invoke these built-in intents *without* using your skill's invocation name (see [below](#)). If your skill is currently playing audio, or was the skill most recently playing audio, these intents are automatically sent to your skill. Your code needs to expect them and not return an error.

If any of these intents does not apply to your skill, handle it in a graceful way in your code. For instance, you could return a response with text-to-speech indicating that the command is not relevant to the skill. The specific message depends on the skill and whether the intent is one that might make sense at some point, for example:

- For a podcast skill, the `AMAZON.ShuffleOnIntent` intent might return the message: "I can't shuffle a podcast."
- For version 1.0 of a music skill that doesn't yet support playlists and shuffling, the `AMAZON.ShuffleOnIntent` intent might return: "Sorry, I can't shuffle music yet."

**❶ Note:** If your skill uses the `AudioPlayer` directives, you cannot extend the above built-in intents with your own sample utterances.

### Invoking the Built-in Intents for Playback Control

When your skill sends a `Play` directive to begin playback, the Alexa service sends the audio stream to the device for playback. Once the session ends normally (for instance, if your response included the `shouldEndSession` flag set to `true`), Alexa remembers that your skill started the playback *until* the user does one of the following:

- Invokes audio playback with a different skill.
- Invokes another service that streams audio, such as the built-in music service or the `flash briefing`.
- Reboots the device.

During this time, users can invoke the following built-in playback control intents without using your skill's invocation name:

- AMAZON.CancelIntent
- AMAZON.LoopOffIntent
- AMAZON.LoopOnIntent
- AMAZON.NextIntent
- AMAZON.PauseIntent
- AMAZON.PreviousIntent
- AMAZON.RepeatIntent
- AMAZON.ResumeIntent
- AMAZON.ShuffleOffIntent
- AMAZON.ShuffleOnIntent
- AMAZON.StartOverIntent



sample utterance "play the latest episode."

**User:** Alexa, ask My Podcast Player to play the latest episode.

Alexa opens a new skill session and sends the My Podcast Player skill the normal PlayLatestEpisode. My Podcast Player sends a Play directive. The skill session closes and audio begins playing.

**User:** Alexa, next. (note no invocation name used.)

Alexa opens a new skill session and sends the My Podcast Player skill AMAZON.NextIntent.

My Podcast Player takes appropriate action for 'next' and closes the skill session.

**User:** Alexa, pause. (again, no invocation name.)

Alexa opens a new skill session and sends the skill AMAZON.PauseIntent.

My Podcast Player sends a Stop directive and closes the skill session. The audio is stopped.

Although at this point the audio is not playing and there is no current session, the Alexa service is still tracking My Podcast Player as the skill that most recently streamed audio. Assuming the device remains on and the user does not use any other audio streaming skills or services, the following could take place at any time later:

**User:** Alexa, resume. (note no invocation name used.)

Alexa opens a new skill session and sends My Podcast Player the AMAZON.ResumeIntent.

My Podcast Player takes appropriate action to determine the previously playing track and send a new Play directive to restart playback.

This only applies to the built-in intents. The intents you define (such as the example PlayLatestEpisode intent) must be invoked using a normal invocation phrase.

**Note:** In the above scenario, when your skill is not in an active session but is playing audio, or was the skill most recently playing audio, utterances such as 'stop' send your skill an AMAZON.PauseIntent instead of an AMAZON.StopIntent.

## AudioPlayer Directives

AudioPlayer provides the following directives:

Directive	Description
AudioPlayer.Play	Sends Alexa a command to stream the audio file identified by the specified audioItem.
AudioPlayer.Stop	Stops any currently playing audio stream.
AudioPlayer.ClearQueue	Clears the queue of all audio streams.

When including a directive in your response, set the type property to the directive you want to send. Include directives in the directives array in your response:

```
{
  "version": "1.0",
  "sessionAttributes": {},
```



```
{
  "type": "AudioPlayer.Play",
  "playBehavior": "string",
  "audioItem": {
    "stream": {
      "token": "string",
      "url": "string",
      "offsetInMilliseconds": 0
    }
  }
},
"shouldEndSession": true
}
```

For the full response format, see [Response Format](#) in the [JSON Interface Reference for Custom Skills](#).

**Tip:** When responding to a [LaunchRequest or IntentRequest](#), your response can include both AudioPlayer directives and [standard response properties](#) such as `outputSpeech`, `card`, and `reprompt`. For example, if you provide `outputSpeech` in the same response as an `Play` directive, Alexa speaks the provided text before beginning to stream the audio. Note that the rules are different when responding to [AudioPlayer requests](#).

## AudioPlayer Requests

AudioPlayer sends the following requests to notify your skill about changes to the playback state:

Request Type	Description
<a href="#">AudioPlayer.PlaybackStarted</a>	Sent when Alexa begins playing the audio stream previously sent in a <code>Play</code> directive. This lets your skill verify that playback began successfully.
<a href="#">AudioPlayer.PlaybackFinished</a>	Sent when the stream Alexa is playing comes to an end on its own.
<a href="#">AudioPlayer.PlaybackStopped</a>	Sent when Alexa stops playing an audio stream in response to a voice request or an AudioPlayer directive.
<a href="#">AudioPlayer.PlaybackNearlyFinished</a>	Sent when the currently playing stream is nearly complete and the device is ready to receive a new stream.
<a href="#">AudioPlayer.PlaybackFailed</a>	Sent when Alexa encounters an error when attempting to play a stream.

**Important:** These requests do not include the `session` object, since they are not sent in the context of a skill session. Use the `context` object to get details such as the `applicationId` and `userId`.

```
{
  "version": "string",
  "context": {
    "System": {
      "application": {},
      "user": {},
      "device": {}
    }
  },
  "request": {
    "type": "AudioPlayer.PlaybackStarted",
    "payload": {
      "playBehavior": "string"
    }
  }
}
```



```
}
```

For the full request format, see [Request Format](#) in the [JSON Interface Reference for Custom Skills](#).

**Note:** When responding to AudioPlayer requests, you can only respond with [AudioPlayer directives](#). The response cannot include any of the standard properties such as `outputSpeech`. In addition, some requests limit the directives you can use, such as not allowing `Play`. Sending a response with unsupported properties causes an error. See the request types below for the limits on each request.

Also note that your service is not required to return a response to the AudioPlayer requests.

## Play Directive

Sends Alexa a command to stream the audio file identified by the specified `audioItem`. Use the `playBehavior` parameter to determine whether the stream begins playing immediately, or is added to the queue.

**Note:** You can only send one `Play` directive in a request.

When sending a `Play` directive, you normally set the `shouldEndSession` flag in the [response object](#) to `true` to end the session. If you set this flag to `false`, Alexa sends the stream to the device for playback, then immediately pauses the stream to listen for the user's response.

```
{
  "type": "AudioPlayer.Play",
  "playBehavior": "string",
  "audioItem": {
    "stream": {
      "url": "string",
      "token": "string",
      "expectedPreviousToken": "string",
      "offsetInMilliseconds": 0
    }
  }
}
```

Parameter	Description	Type	Required
<code>type</code>	Set to <code>AudioPlayer.Play</code> .	string	yes

HOME

ALEXA

SERVICES &amp; APIs

DEVICES

RESOURCES

BLOG

SUPPORT

Search...



**playBehavior** Describes playback behavior. Accepted values: stri ng

- **REPLACE\_ALL**: Immediately begin playback of the specified stream, and replace current and enqueued streams.
- **ENQUEUE**: Add the specified stream to the end of the current queue. This does not impact the currently playing stream.
- **REPLACE\_ENQUEUED**: Replace all streams in the queue. This does not impact the currently playing stream.

---

**audioItem** Contains an object providing information about the audio stream to play. obje ct

---

**audioItem.stream** Contains an object representing the audio stream to play. obje ct

---

**audioItem.stream.url** Identifies the location of audio content at a remote HTTPS location. stri ng

The audio file must be hosted at an Internet-accessible HTTPS endpoint. HTTPS is required, and the domain hosting the files must present a valid, trusted SSL certificate. Self-signed certificates cannot be used. Many content hosting services provide this. For example, you could host your files at a service such as [Amazon Simple Storage Service \(Amazon S3\)](#) (an [Amazon Web Services](#) offering).

The supported formats for the audio file include AAC/MP4, MP3, HLS, PLS and M3U. Bitrates: 16kbps to 384 kbps.

---

**audioItem.stream.token** An opaque token that represents the audio stream. stri ng  
This token cannot exceed 1024 characters.



This property is required and allowed only when the `playBehavior` is `ENQUEUE`. This is used to prevent potential race conditions if requests to progress through a playlist and change tracks occur at the same time. For example:

1. The skill is streaming *track 2* in a playlist of several tracks.
2. The user says "Alexa, go back," which sends an `AMAZON.PreviousIntent`.
3. At about the same time, track 2 is nearly finished, so Alexa sends a `PlaybackNearlyFinished` request.
4. The skill handles the `AMAZON.PreviousIntent` first and sends a new `Play` directive with track 1. This track begins playing. The already-sent `PlaybackNearlyFinished` request is now outdated, since it assumed that track 2 was playing.
5. The skill handles the now-outdated `PlaybackNearlyFinished` request and sends a `Play` directive with *track 3*, since this is the next track after the originally playing track 2. This request includes `expectedPreviousToken` set to *track 2*.
6. The `expectedPreviousToken` provided in the directive does not match the token for the actively playing stream, so the device ignores this directive.



ished request. The skill responds with a Play directive for *track 2*. This track begins playing once track 1 finishes.

If this check was not in place, the directive sent in step 5 would put *track 3* on the queue, which would cause the audio to skip from track 1 to track 3 when track 1 finishes.

**Note:** Including this property when `playBehavior` is any other value (`REPLACE_ALL` or `REPLACE_ENQUEUED`) causes an error.

<code>audioItem.stream.offsetInMilliseconds</code>	The timestamp in the stream from which Alexa should begin playback. Set to 0 to start playing the stream from the beginning. Set to any other value to start playback from that associated point in the stream.	long	yes
--	---	------	-----

## Stop Directive

Stops the current audio playback.

```
{
  "type": "AudioPlayer.Stop"
}
```

Parameter	Description	Type	Required
<code>type</code>	Set to <code>AudioPlayer.Stop</code>	string	yes

## ClearQueue Directive

Clears the audio playback queue. You can set this directive to clear the queue without stopping the currently playing stream, or clear the queue and stop any currently playing stream.

```
{
  "type": "AudioPlayer.ClearQueue",
  "clearBehavior" : "string"
}
```

Parameter	Description	Type	Required
-----------	-------------	------	----------



clearBehavior	Describes the clear queue behavior. Accepted values:	string	yes
		■ CLEAR_ENQUEUED : clears the queue and continues to play the currently playing stream	
		■ CLEAR_ALL : clears the entire playback queue and stops the currently playing stream (if applicable).	

## PlaybackStarted Request

Sent when Alexa begins playing the audio stream previously sent in a Play directive. This lets your skill verify that playback began successfully.

This request is also sent when Alexa resumes playback after pausing it for a voice request.

```
{
  "type": "AudioPlayer.PlaybackStarted",
  "requestId": "string",
  "timestamp": "string",
  "token": "string",
  "offsetInMilliseconds": 0,
  "locale": "string"
}
```

### Parameters

Parameter	Description	Type
type	AudioPlayer.PlaybackStarted	string
requestId	Represents a unique identifier for the specific request.	string
timestamp	Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to <a href="#">verify the request when hosting your skill as a web service</a> .	string
token	An opaque token that represents the audio stream. You provide this token when sending the Play directive.	string
offsetInMilliseconds	Identifies a track's offset in milliseconds when the PlaybackStarted request is sent.	long
locale	A string indicating the user's locale. For example: en-US.	string



## Valid Response Types

- Your skill can respond to `PlaybackStarted` with a [Stop](#) or [ClearQueue](#) directive.
- The response cannot include any standard properties such as `outputSpeech`, `card`, or `reprompt`.

## PlaybackFinished Request

Sent when the stream Alexa is playing comes to an end on its own.

**Note:** If your skill explicitly stops the playback with the [Stop](#) directive, Alexa sends `PlaybackStopped` instead of `PlaybackFinished`.

```
{
  "type": "AudioPlayer.PlaybackFinished",
  "requestId": "string",
  "timestamp": "string",
  "token": "string",
  "offsetInMilliseconds": 0,
  "locale": "string"
}
```

## Parameters

Parameter	Description	Type
type	AudioPlayer.PlaybackFinished	string
requestId	Represents a unique identifier for the specific request.	string
timestamp	Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to <a href="#">verify the request when hosting your skill as a web service</a> .	string
token	An opaque token that represents the audio stream. You provide this token when sending the <code>Play</code> directive.	string
offsetInMilliseconds	Identifies a track's offset in milliseconds when the <code>PlaybackFinished</code> request is sent.	long
locale	A string indicating the user's locale. For example: en-US.	string

## Valid Response Types

- Your skill can respond to `PlaybackFinished` with a [Stop](#) or [ClearQueue](#) directive.



## PlaybackStopped Request

Sent when Alexa stops playing an audio stream in response to one of the following AudioPlayer directives:

- Stop
- Play with a playBehavior of REPLACE\_ALL .
- ClearQueue with a clearBehavior of CLEAR\_ALL .

This request is also sent if the user makes a voice request to Alexa, since this temporarily pauses the playback. In this case, the playback begins automatically once the voice interaction is complete.

**Note:** If playback stops because the audio stream comes to an end on its own, Alexa sends PlaybackFinished instead of PlaybackStopped.

```
{
  "type": "AudioPlayer.PlaybackStopped",
  "requestId": "string",
  "timestamp": "string",
  "token": "string",
  "offsetInMilliseconds": 0,
  "locale": "string"
}
```

### Parameters

Parameter	Description	Type
type	AudioPlayer.PlaybackStopped	string
requestId	Represents a unique identifier for the specific request.	string
timestamp	Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to verify the request when hosting your skill as a web service.	string
token	An opaque token that represents the audio stream. You provide this token when sending the Play directive.	string
offsetInMilliseconds	Identifies a track's offset in milliseconds when the PlaybackStopped request is sent.	long
locale	A string indicating the user's locale. For example: en-US .	string

### Valid Response Types

Your skill cannot return a response to PlaybackStopped .



Sent when the device is ready to add the next stream to the queue.

To progress through a playlist of audio streams, respond to this request with a [Play](#) directive for the next stream and set `playBehavior` to `ENQUEUE` or `REPLACE_ENQUEUED`. This adds the new stream to the queue without stopping the current playback. Alexa begins streaming the new audio item once the currently playing track finishes.

```
{
  "type": "AudioPlayer.PlaybackNearlyFinished",
  "requestId": "string",
  "timestamp": "string",
  "token": "string",
  "offsetInMilliseconds": 0,
  "locale": "string"
}
```

### Parameters

Parameter	Description	Type
type	AudioPlayer.PlaybackNearlyFinished	string
requestId	Represents a unique identifier for the specific request.	string
timestamp	Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to <a href="#">verify the request when hosting your skill as a web service</a> .	string
token	An opaque token that represents the audio stream that is currently playing. You provide this token when sending the <code>Play</code> directive.	string
offsetInMilliseconds	Identifies a track's offset in milliseconds when the <code>PlaybackNearlyFinished</code> request is sent.	long
locale	A string indicating the user's locale. For example: <code>en-US</code> .	string

### Valid Response Types

- Your skill can respond to `PlaybackNearlyFinished` with any `AudioPlayer` directive.
- The response cannot include any standard properties such as `outputSpeech`, `card`, or `reprompt`.

## PlaybackFailed Request

Sent when Alexa encounters an error when attempting to play a stream.

```
{
  "type": "AudioPlayer.PlaybackFailed",
```



```

    "type": "string",
    "message": "string"
},
"currentPlaybackState": {
    "token": "string",
    "offsetInMilliseconds": 0,
    "playerActivity": "string"
}
}

```

This request type includes two `token` properties – one as a property of the `request` object, and one as a property of the `currentPlaybackState` object. The `request.token` property represents the stream that failed to play. The `currentPlaybackState.token` property can be different if Alexa is playing a stream and the error occurs when attempting to buffer the next stream on the queue. In this case, `currentPlaybackState.token` represents the stream that was successfully playing.

Parameter	Description	Type
<code>type</code>	<code>AudioPlayer.PlaybackFailed</code>	string
<code>requestId</code>	Represents a unique identifier for the specific request.	string
<code>timestamp</code>	Provides the date and time when Alexa sent the request as an ISO 8601 formatted string. Used to <a href="#">verify the request when hosting your skill as a web service</a> .	string
<code>token</code>	An opaque token provided by the <code>Play</code> directive that represents the stream that failed to play.	string
<code>locale</code>	A string indicating the user's locale. For example: <code>en-US</code> .	string
<code>error</code>	Contains an object with error information	object
<code>error.type</code>	Identifies the specific type of error. The table below provides details for each error type.	string
<code>error.message</code>	A description of the error the device has encountered.	string
<code>currentPlaybackState</code>	Contains an object providing details about the playback activity occurring at the time of the error.	object
<code>currentPlaybackState.token</code>	An opaque token that represents the audio stream currently playing when the error occurred. Note that this may be different from the value of the <code>request.token</code> property.	string



.	offsetInMilliseconds	offset in milliseconds when the error occurred.
.	currentPlaybackState	Identifies the player state when the error occurred: PLAYING, PAUSED, FINISHED, BUFFER_UNDERUN, or IDLE.

Error Type	Description
MEDIA_ERROR_UNKNOWN	An unknown error occurred.
MEDIA_ERROR_INVALID_REQUEST	Alexa recognized the request as being malformed. E.g. bad request, unauthorized, forbidden, not found, etc.
MEDIA_ERROR_SERVICE_UNAVAILABLE	Alexa was unable to reach the URL for the stream.
MEDIA_ERROR_INTERNAL_SERVER_ERR OR	Alexa accepted the request, but was unable to process the request as expected.
MEDIA_ERROR_INTERNAL_DEVICE_ERR OR	There was an internal error on the device.

### Valid Response Types

- Your skill can respond to `PlaybackFailed` with any `AudioPlayer` directive.
- The response cannot include any standard properties such as `outputSpeech`, `card`, or `reprompt`.

## System.ExceptionEncountered Request

If a response to an `AudioPlayer` request causes an error, your skill is sent a `System.ExceptionEncountered` request. Any directives included in the response are ignored.

```
{
  "type": "System.ExceptionEncountered",
  "requestId": "string",
  "timestamp": "string",
  "locale": "string",
  "error": {
    "type": "string",
    "message": "string"
  },
  "cause": {
    "requestId": "string"
  }
}
```

### Parameters

Parameter	Description	Type
type	System.ExceptionEncountered	string
requestId	Represents a unique identifier for the specific request.	string

[HOME](#)[ALEXA](#)[SERVICES & APIs](#)[DEVICES](#)[RESOURCES](#)[BLOG](#)[SUPPORT](#)

Search...



time when Alexa sent the request as an ISO 8601 formatted string. Used to [verify the request when hosting your skill as a web service.](#)

locale	A string indicating the user's locale. For example: en-US.	string
error	Contains an object with error information	object
error.type	Identifies the specific type of error (INVALID_RESPONSE, DEVICE_COMMUNICATION_ERROR, INTERNAL_ERROR).	string
error.message	A description of the error the device has encountered.	string
cause.requestId	The requestId for the request that caused the error	string

### Valid Response Types

Your skill cannot return a response to `System.ExceptionEncountered`.

## Other Reference Materials

- [JSON Interface Reference for Custom Skills](#)
- [Standard Request Type Reference](#)

Interfaces:

- [PlaybackController Interface Reference](#)

[SITEMAP](#)

FOLLOW US



HOME	ALEXA	SERVICES & APIs	DEVICES	RESOURCES	BLOG	SUPPORT	Search...	
Alexa Voice Service	Engage		Amazon Fire TV		Learning Center		#Announcements	
Alexa Fund	Build		Dash Replenishment Service		Development Tools		#AmazonUnderground	
			Fire Phone		Promotional Tools		#Alexa	
			Amazon Echo		Marketing Tips		#AmazonFireTV	
			Amazon Tap		Other Resources		#FireTablets	
							#How To	
							#Monetization	
							#APIs	
							#Marketing	
							#Gamedev	
							#Developer Spotlight	

**Support**[Submitting Your Apps](#)[FAQs](#)[Forums](#)[Contact Us](#)[App Distribution Agreement](#)[Mobile Ad Network Publisher Agreement](#)[Mobile Ad Network Program Participation Requirements](#)[Advertise Your App With Amazon Agreement](#)[Program Materials License Agreement](#)[Trademark Guidelines](#)[Terms of Use](#)

© 2010-2016, Amazon.com, Inc. or its affiliates. All Rights Reserved.